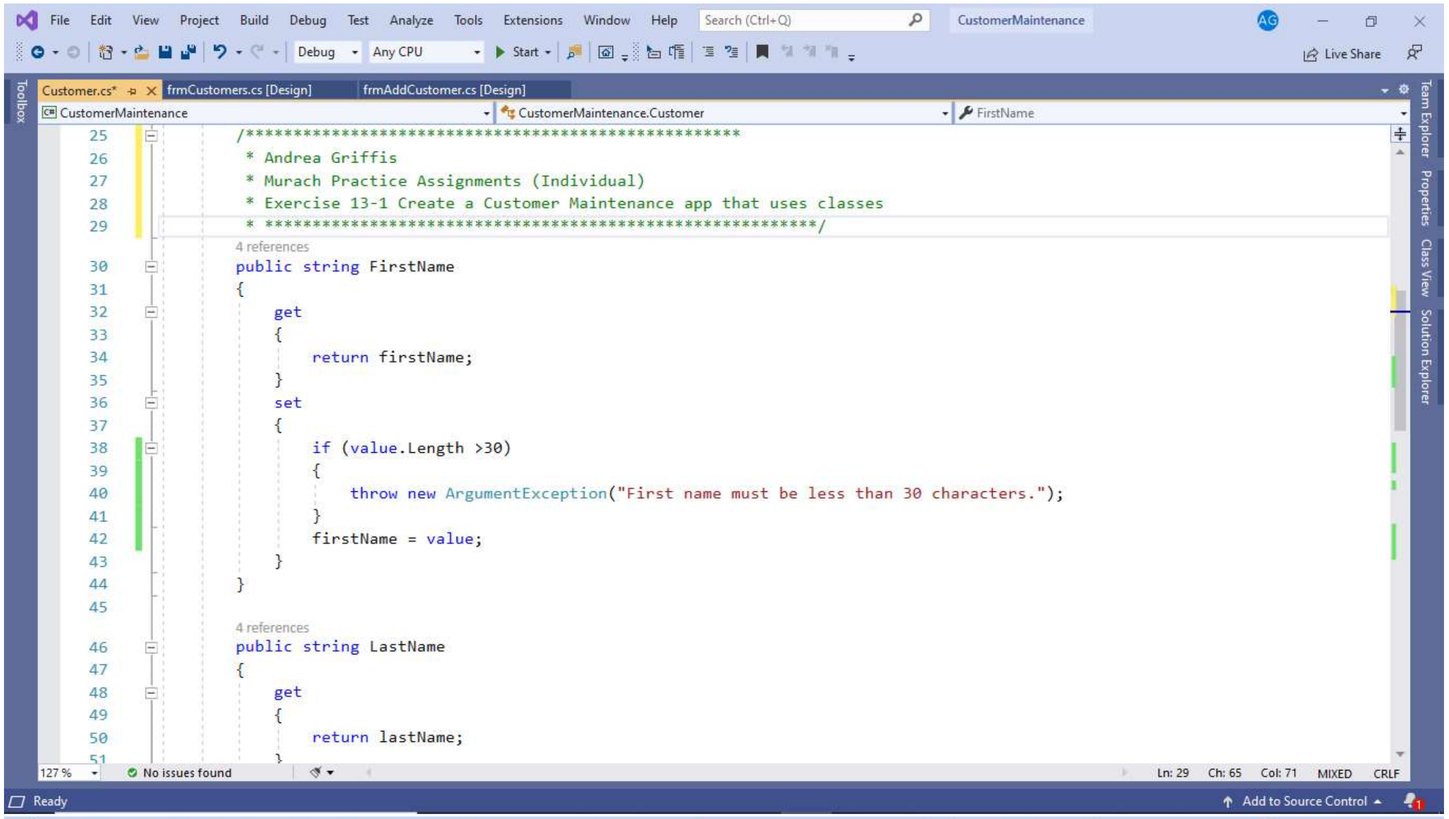
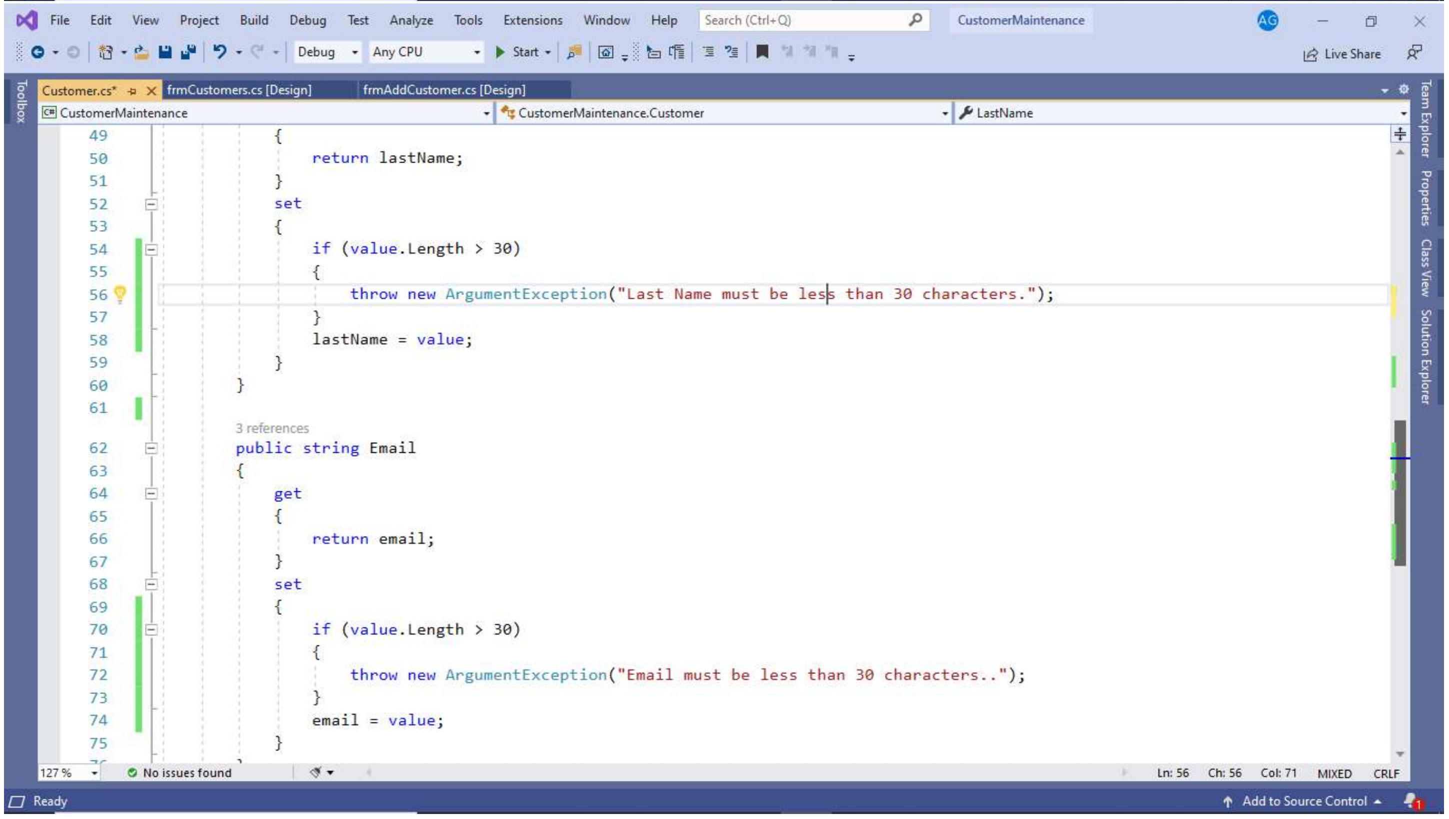
Andrea Griffis
CIS 123 C#
Assignment: Murach Practice Assignments (Individual)

Exercise 13-1 Create a Customer Maintenance app that uses classes (#'s 1-16)

## Open the project and add validation code to the customer class #1-4



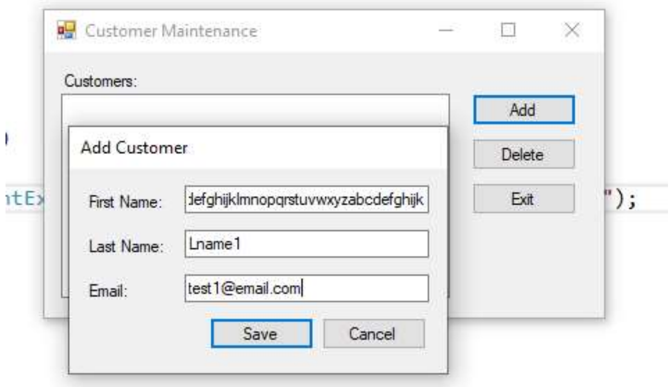
```
/****************************************************
 * Andrea Griffis
 * Murach Practice Assignments (Individual)
 * Exercise 13-1 Create a Customer Maintenance app that uses classes
 * *********************************************************/
4 references
public string FirstName
{
    get
    {
        return firstName;
    }
    set
    {
        if (value.Length >30)
        {
            throw new ArgumentException("First name must be less than 30 characters.");
        }
        firstName = value;
    }
}

4 references
public string LastName
{
    get
    {
        return lastName;
```



```
    {
        return lastName;
    }
    set
    {
        if (value.Length > 30)
        {
            throw new ArgumentException("Last Name must be less than 30 characters.");
        }
        lastName = value;
    }
}

3 references
public string Email
{
    get
    {
        return email;
    }
    set
    {
        if (value.Length > 30)
        {
            throw new ArgumentException("Email must be less than 30 characters..");
        }
        email = value;
    }
```

Test the application for > 30 characters

1) 1st name>30 2) 2nd name >30 3) Email >30


Customer Maintenance

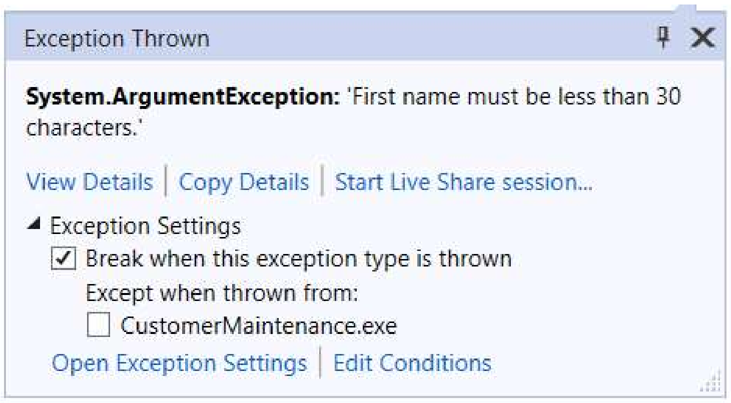
Customers:

Add Customer

First Name: defghijklmnopqrstuvwxyzabcdefghijk

Last Name: Lname1

Email: test1@email.com


Exception Thrown

**System.ArgumentException:** 'First name must be less than 30 characters.'

View Details | Copy Details | Start Live Share session...

Exception Settings

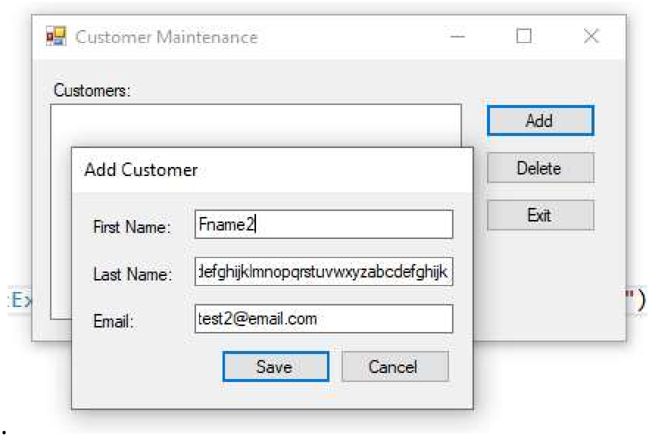
[✓] Break when this exception type is thrown

Except when thrown from:

[ ] CustomerMaintenance.exe

Open Exception Settings | Edit Conditions

1.


Customer Maintenance

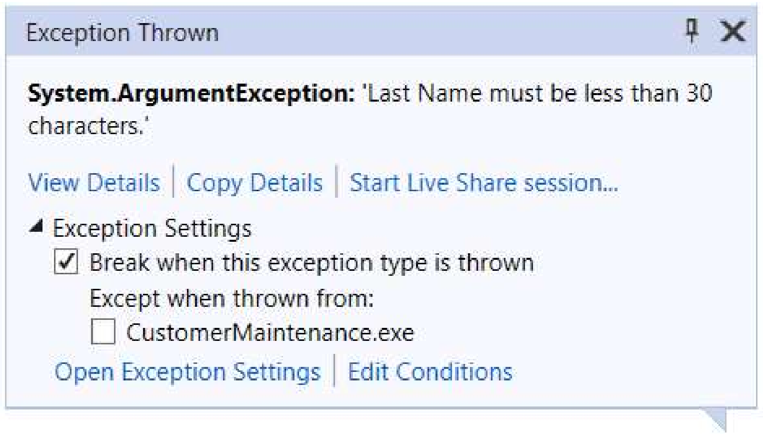
Customers:

Add Customer

First Name: Fname2

Last Name: defghijklmnopqrstuvwxyzabcdefghijk

Email: test2@email.com


Exception Thrown

**System.ArgumentException:** 'Last Name must be less than 30 characters.'

View Details | Copy Details | Start Live Share session...

Exception Settings

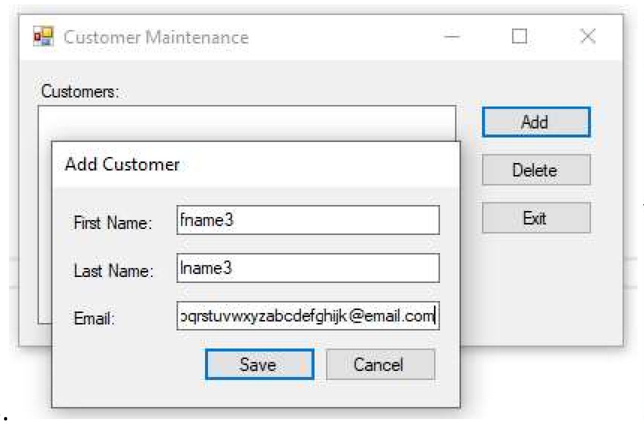
[✓] Break when this exception type is thrown

Except when thrown from:

[ ] CustomerMaintenance.exe

Open Exception Settings | Edit Conditions

2.


Customer Maintenance

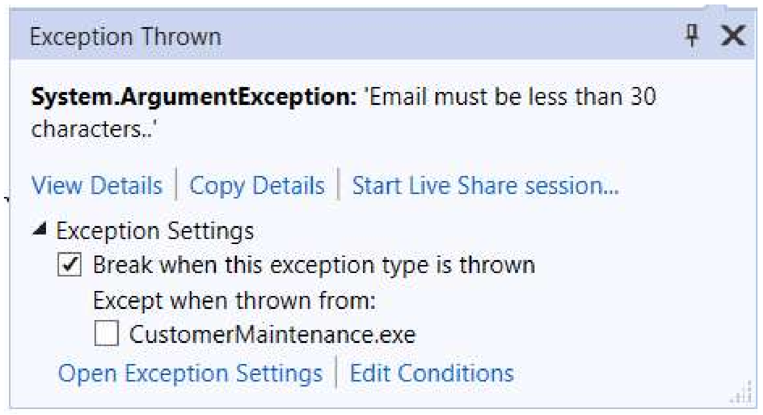
Customers:

Add Customer

First Name: fname3

Last Name: lname3

Email: pqrstuvwxyzabcdefghijk@email.com


Exception Thrown

**System.ArgumentException:** 'Email must be less than 30 characters..'

View Details | Copy Details | Start Live Share session...

Exception Settings

[✓] Break when this exception type is thrown

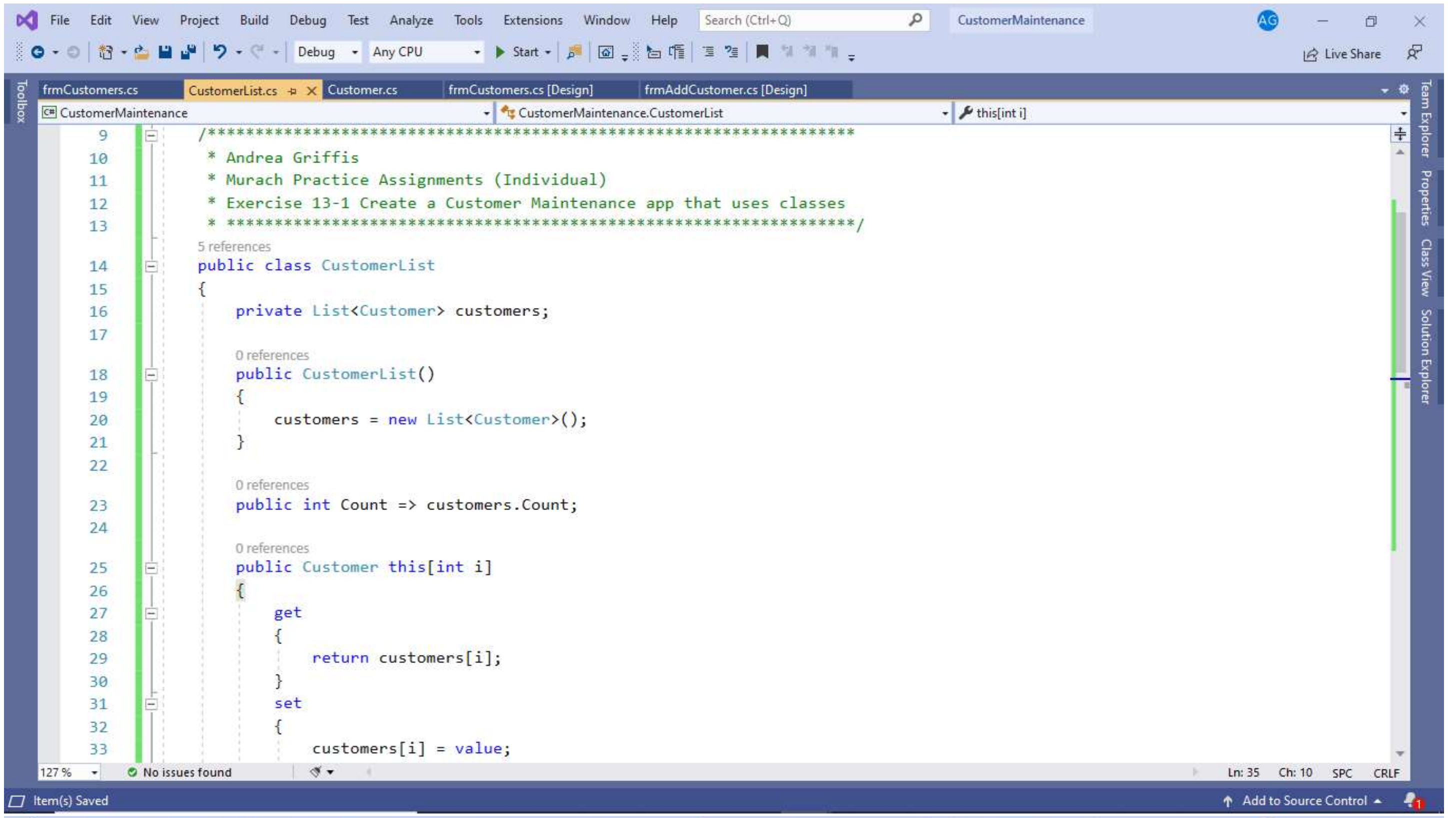
Except when thrown from:

[ ] CustomerMaintenance.exe

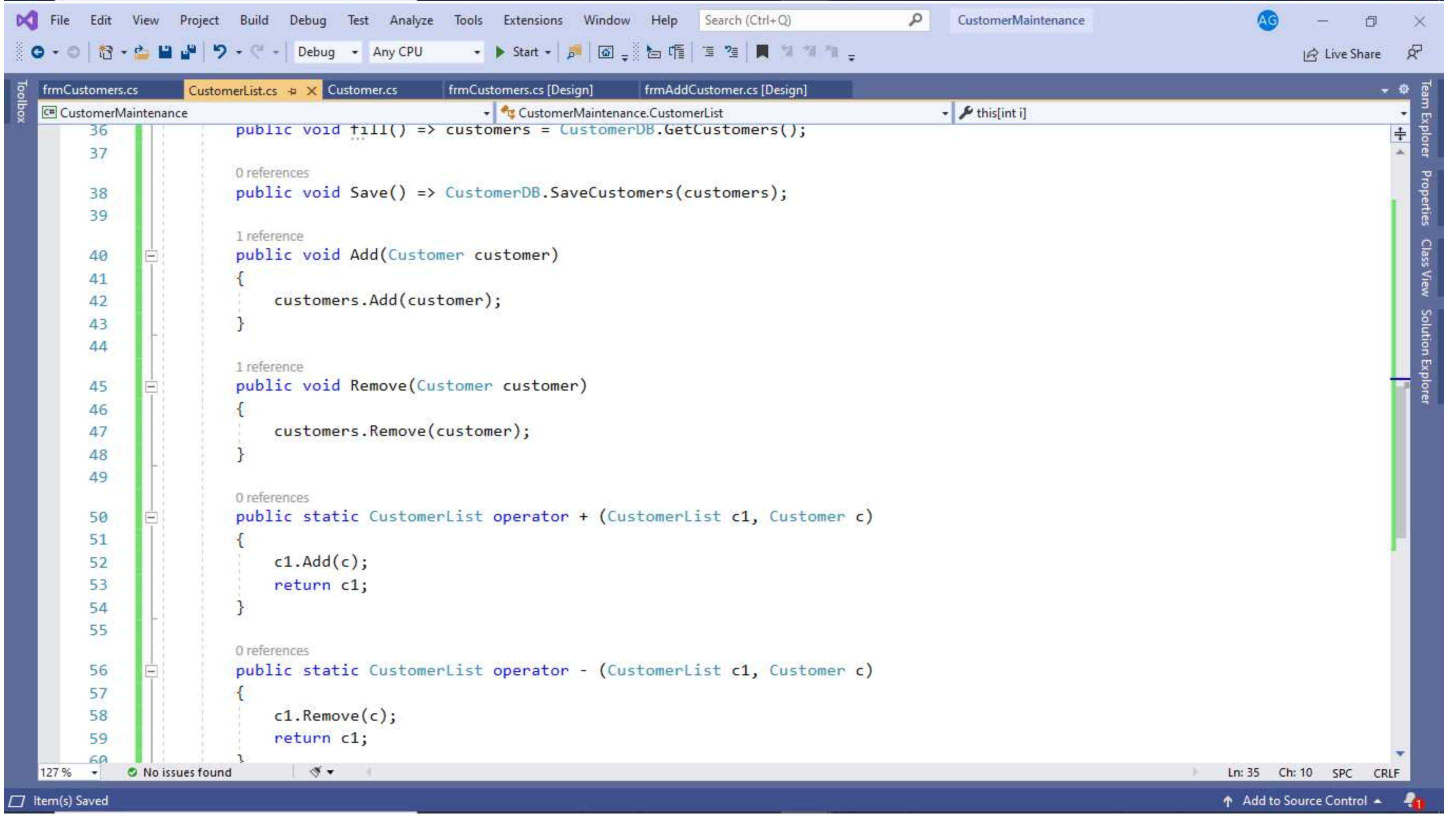
Open Exception Settings | Edit Conditions

3.

# Add a CustomerList class #5-8

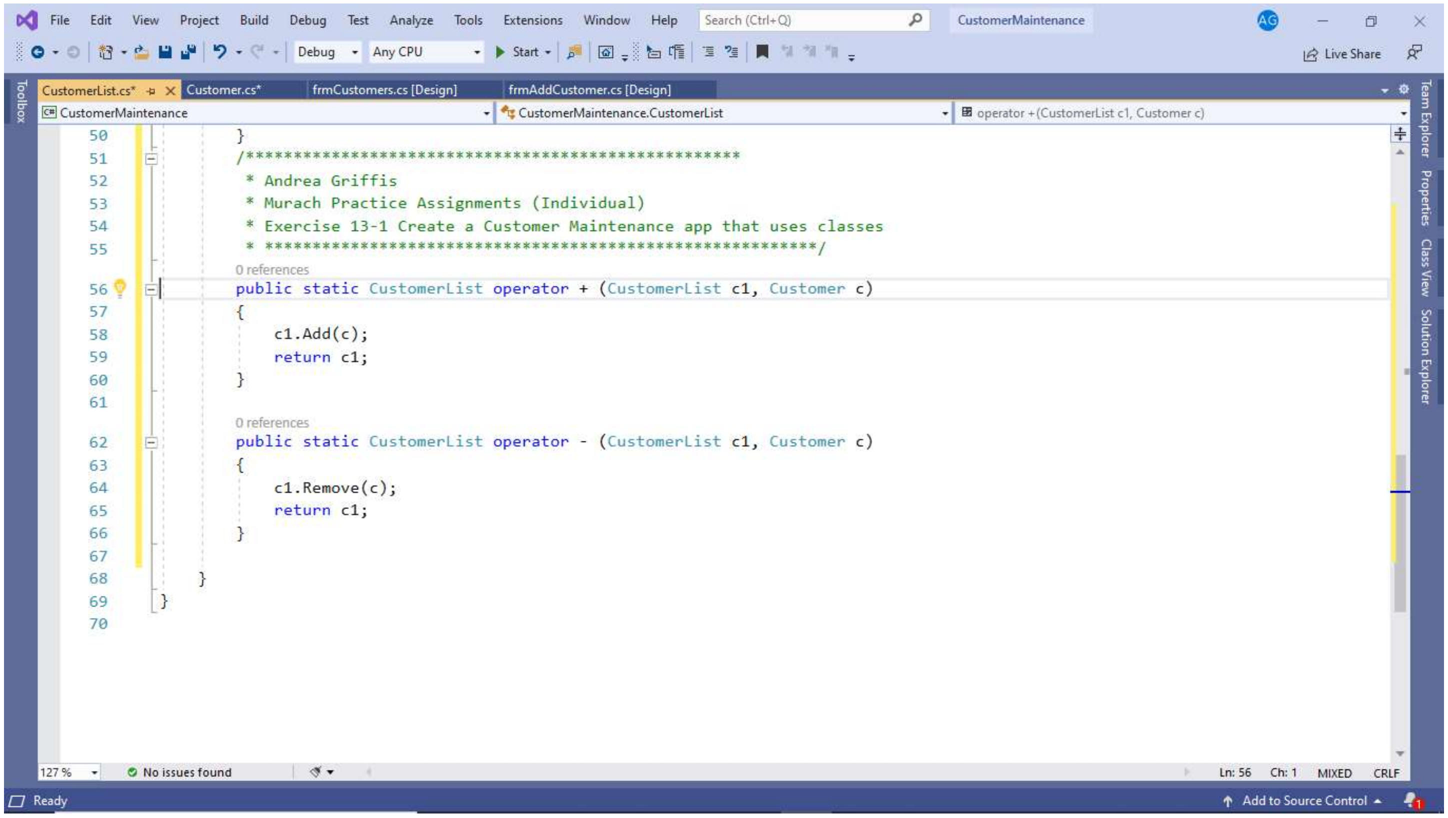


```
/*******************************************************************
 * Andrea Griffis
 * Murach Practice Assignments (Individual)
 * Exercise 13-1 Create a Customer Maintenance app that uses classes
 * *******************************************************************/
5 references
public class CustomerList
{
    private List<Customer> customers;

    0 references
    public CustomerList()
    {
        customers = new List<Customer>();
    }

    0 references
    public int Count => customers.Count;

    0 references
    public Customer this[int i]
    {
        get
        {
            return customers[i];
        }
        set
        {
            customers[i] = value;
```



```
    public void Fill() => customers = CustomerDB.GetCustomers();

    0 references
    public void Save() => CustomerDB.SaveCustomers(customers);

    1 reference
    public void Add(Customer customer)
    {
        customers.Add(customer);
    }

    1 reference
    public void Remove(Customer customer)
    {
        customers.Remove(customer);
    }

    0 references
    public static CustomerList operator + (CustomerList c1, Customer c)
    {
        c1.Add(c);
        return c1;
    }

    0 references
    public static CustomerList operator - (CustomerList c1, Customer c)
    {
        c1.Remove(c);
        return c1;
```

Add overloaded operators to the CustomerList class #9-10

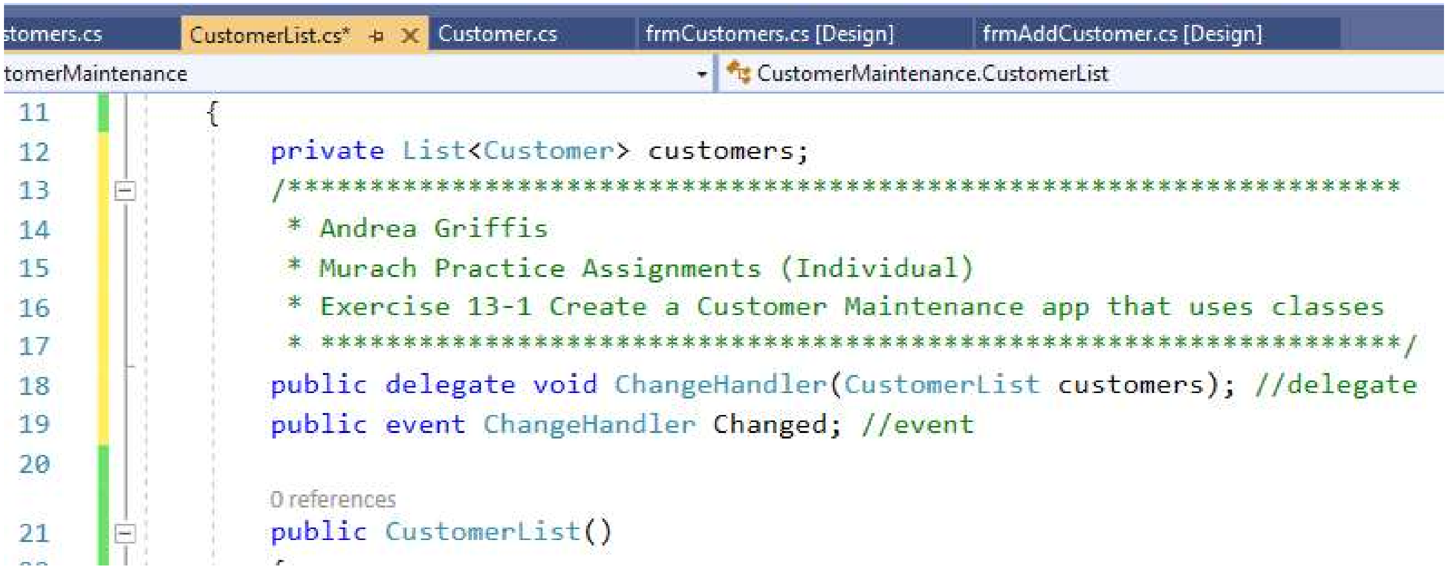


CustomerList.cs*
Customer.cs*
frmCustomers.cs [Design]
frmAddCustomer.cs [Design]
CustomerMaintenance
CustomerMaintenance.CustomerList
operator +(CustomerList c1, Customer c)

```
        }
        /****************************************************
         * Andrea Griffis
         * Murach Practice Assignments (Individual)
         * Exercise 13-1 Create a Customer Maintenance app that uses classes
         * ********************************************************/
        0 references
        public static CustomerList operator + (CustomerList c1, Customer c)
        {
            c1.Add(c);
            return c1;
        }

        0 references
        public static CustomerList operator - (CustomerList c1, Customer c)
        {
            c1.Remove(c);
            return c1;
        }

    }
}
```

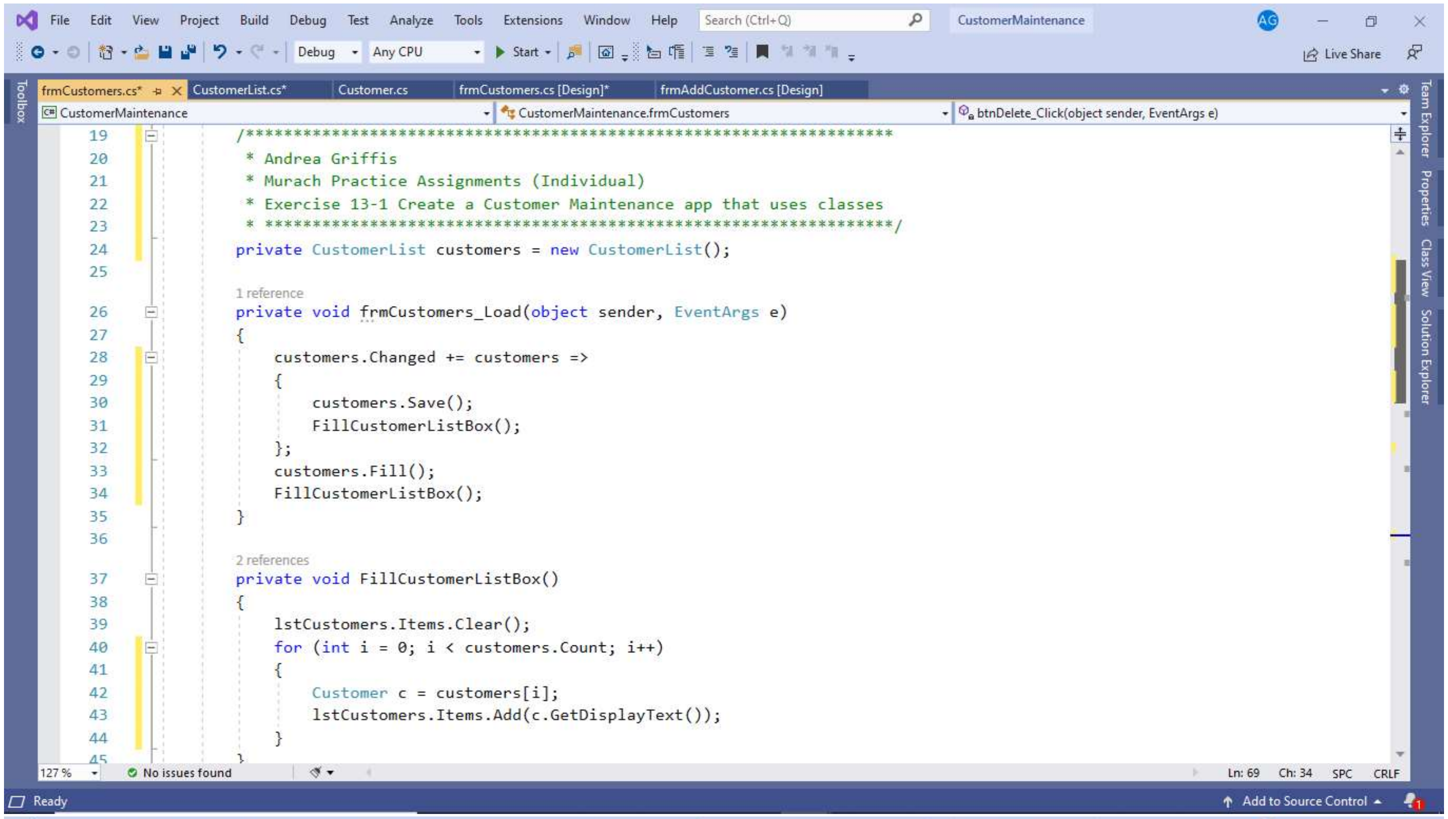
Add a delegate and an event to the CustomerList class #12-16

(11 & 12)



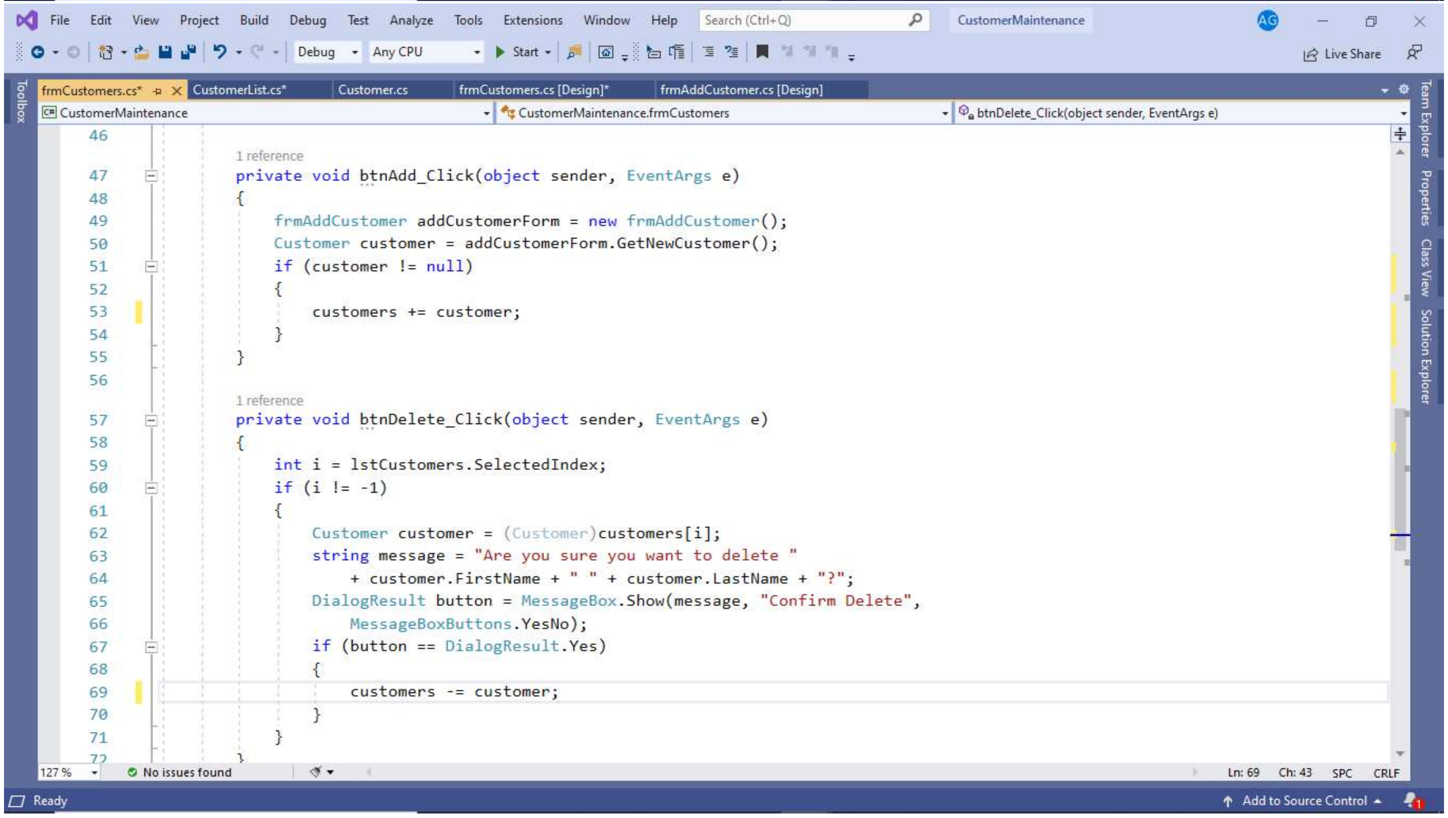
```
    {
        private List<Customer> customers;
        /*******************************************************************
         * Andrea Griffis
         * Murach Practice Assignments (Individual)
         * Exercise 13-1 Create a Customer Maintenance app that uses classes
         * *****************************************************************/
        public delegate void ChangeHandler(CustomerList customers); //delegate
        public event ChangeHandler Changed; //event

        0 references
        public CustomerList()
```

(13-15)



File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
CustomerMaintenance
frmCustomers.cs* CustomerList.cs* Customer.cs frmCustomers.cs [Design]* frmAddCustomer.cs [Design]
CustomerMaintenance.frmCustomers
btnDelete_Click(object sender, EventArgs e)
No issues found
Ln: 69 Ch: 34 SPC CRLF
Ready

```
/*********************************************************************
 * Andrea Griffis
 * Murach Practice Assignments (Individual)
 * Exercise 13-1 Create a Customer Maintenance app that uses classes
 * ********************************************************************/
private CustomerList customers = new CustomerList();

private void frmCustomers_Load(object sender, EventArgs e)
{
    customers.Changed += customers =>
    {
        customers.Save();
        FillCustomerListBox();
    };
    customers.Fill();
    FillCustomerListBox();
}

private void FillCustomerListBox()
{
    lstCustomers.Items.Clear();
    for (int i = 0; i < customers.Count; i++)
    {
        Customer c = customers[i];
        lstCustomers.Items.Add(c.GetDisplayText());
    }
```



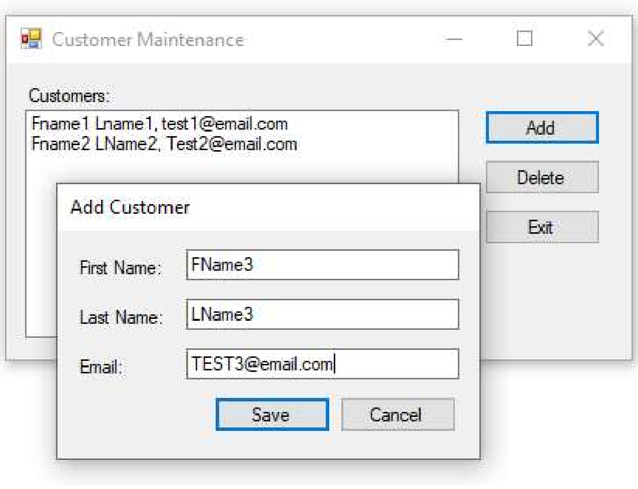
File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
CustomerMaintenance
frmCustomers.cs* CustomerList.cs* Customer.cs frmCustomers.cs [Design]* frmAddCustomer.cs [Design]
CustomerMaintenance.frmCustomers
btnDelete_Click(object sender, EventArgs e)
No issues found
Ln: 69 Ch: 43 SPC CRLF
Ready

```
private void btnAdd_Click(object sender, EventArgs e)
{
    frmAddCustomer addCustomerForm = new frmAddCustomer();
    Customer customer = addCustomerForm.GetNewCustomer();
    if (customer != null)
    {
        customers += customer;
    }
}

private void btnDelete_Click(object sender, EventArgs e)
{
    int i = lstCustomers.SelectedIndex;
    if (i != -1)
    {
        Customer customer = (Customer)customers[i];
        string message = "Are you sure you want to delete "
            + customer.FirstName + " " + customer.LastName + "?";
        DialogResult button = MessageBox.Show(message, "Confirm Delete",
            MessageBoxButtons.YesNo);
        if (button == DialogResult.Yes)
        {
            customers -= customer;
        }
    }
```

## #16 Test the application

1) Add customers 2) Choose Customer to delete

3) Confirm Delete 4) Completed Delete


Customer Maintenance

Customers:

Fname1 Lname1, test1@email.com
Fname2 LName2, Test2@email.com

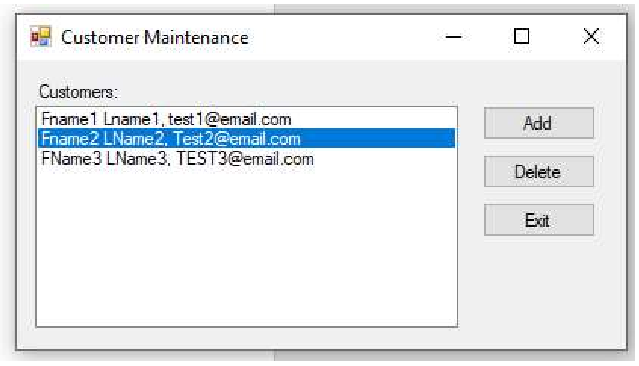
Add Customer

First Name: FName3

Last Name: LName3

Email: TEST3@email.com

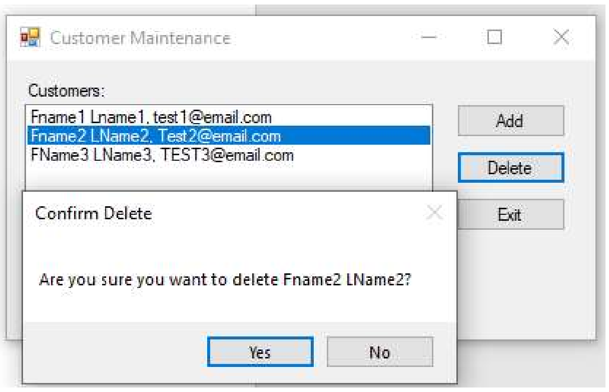
1.


Customer Maintenance

Customers:

Fname1 Lname1, test1@email.com
Fname2 LName2, Test2@email.com
FName3 LName3, TEST3@email.com

2.


Customer Maintenance

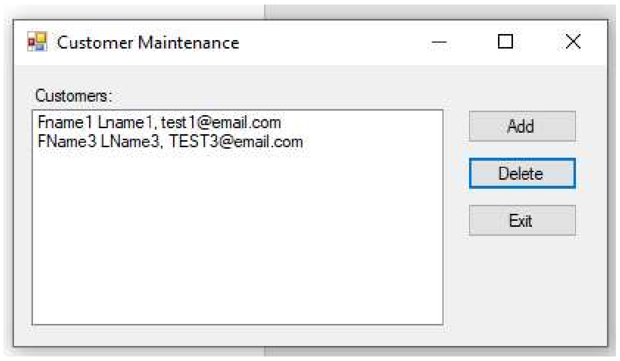
Customers:

Fname1 Lname1, test1@email.com
Fname2 LName2, Test2@email.com
FName3 LName3, TEST3@email.com

Confirm Delete

Are you sure you want to delete Fname2 LName2?

3.


Customer Maintenance

Customers:

Fname1 Lname1, test1@email.com
FName3 LName3, TEST3@email.com

4.